Análisis y Propuesta de Seguridad para SecureWeb Ltd.

1. Objetivo del Ejercicio

El objetivo de este informe es analizar, clasificar y proponer soluciones para cinco eventos de seguridad en la empresa ficticia SecureWeb Ltd., determinando la prioridad de acción para cada uno, en el contexto de una evaluación de seguridad.

2. Escenario

SecureWeb Ltd. es una empresa tecnológica que ha experimentado varios incidentes y riesgos de seguridad. El equipo de seguridad ha documentado cinco eventos y requiere un informe técnico que clasifique cada evento, proponga medidas de mitigación y prevención, y determine una prioridad de acción clara para resolverlos de manera efectiva.

3. Clasificación de Eventos (parte 1)

A continuación, se clasifica cada uno de los cinco eventos de seguridad, determinando si requieren mitigación, prevención, o ambos.

1. **Múltiples intentos fallidos de login desde la misma IP en un corto período de tiempo**.
    - **Clasificación:** Ambos: requiere mitigar y prevenir.
    - **Justificación:** Es un evento en curso que debe ser mitigado (detener el ataque de fuerza bruta), pero también es un riesgo que pudo haberse prevenido con medidas de seguridad robustas desde el diseño del sistema.
2. **Inyección de código JavaScript en los campos de comentarios del blog, que se ejecuta al ser visualizado por otros usuarios**.
    - **Clasificación:** Ambos: requiere mitigar y prevenir.
    - **Justificación:** El incidente ya ocurrió, por lo que se debe mitigar (eliminar el script inyectado), pero la vulnerabilidad de XSS es un riesgo que se pudo haber prevenido con buenas prácticas de desarrollo seguro.
3. **Acceso no autorizado al endpoint /admin/export sin validación de rol**.
    - **Clasificación:** Ambos: requiere mitigar y prevenir.
    - **Justificación:** Es un incidente en curso que requiere mitigar (revocar el acceso y cerrar la sesión), pero la falta de un control de acceso adecuado es una falla de diseño que pudo haberse prevenido.

4. **Uso de una librería JavaScript con una vulnerabilidad conocida no parcheada desde hace 3 meses**.
    - **Clasificación:** Riesgo prevenible por medidas previas (Prevención).
    - **Justificación:** Todavía no ha ocurrido un incidente de seguridad, pero la vulnerabilidad activa es un riesgo que puede ser explotado en cualquier momento. La solución es una medida de prevención (actualizar la librería).
5. **Fallo en el backend que expone mensajes de error con estructura de base de datos al público**.
    - **Clasificación:** Ambos: requiere mitigar y prevenir.
    - **Justificación:** Es un incidente de fuga de información que debe ser mitigado (ocultar los mensajes de error). La causa raíz es una mala práctica de desarrollo que debió ser prevenida desde la fase de codificación.

4. Propuesta Técnica por Evento (parte 2)

A continuación, se proponen medidas de mitigación y prevención específicas para cada uno de los eventos.

1. **Múltiples intentos fallidos de login**
    - **Mitigación:** Implementar un **bloqueo temporal de la IP atacante** (de 10 a 15 minutos) después de un número específico de intentos fallidos (por ejemplo, 5 intentos).
    - **Prevención:** Añadir un **límite de velocidad (rate limiting)** para todas las solicitudes al endpoint de login y habilitar un mecanismo de detección de bots como **reCAPTCHA**.
2. **Inyección de código JavaScript (XSS)**
    - **Mitigación:** Eliminar de forma inmediata el código malicioso de la base de datos o de donde se haya almacenado.
    - **Prevención:** Implementar una **sanitización de entradas** en el backend para limpiar el código HTML y JavaScript antes de almacenarlo en la base de datos. Además, usar el **escapado de salida** para codificar caracteres especiales antes de mostrar el contenido en el navegador.

3. **Acceso no autorizado al endpoint /admin/export**
    - **Mitigación:** Revocar el acceso de inmediato a la cuenta que ha accedido sin autorización.
    - **Prevención:** Implementar un **control de acceso basado en roles (RBAC)**, asegurando que solo los usuarios con el rol de administrador puedan acceder a dicho endpoint.
4. **Librería JavaScript con vulnerabilidad conocida**
    - **Mitigación:** No aplica, ya que no es un incidente activo.
    - **Prevención:** Actualizar la librería a una versión parcheada y segura. Implementar una herramienta de análisis de dependencias (**SAST**) en el ciclo de integración continua para detectar automáticamente vulnerabilidades.
5. **Fallo en el backend que expone errores de BD**
    - **Mitigación:** Deshabilitar los mensajes de error detallados en el entorno de producción y configurar un manejador de errores genérico.
    - **Prevención:** Implementar un **manejo de errores** personalizado y robusto para asegurar que los mensajes de error mostrados al público no revelen información interna del sistema.

5. Prioridad de Acción (parte 3)

Aquí se presenta la lista de los 5 eventos ordenados de mayor a menor prioridad para su intervención.

1. **Acceso no autorizado al endpoint /admin/export**.
    - **Justificación: Máxima prioridad**. Un acceso no autorizado a información sensible de la empresa es un incidente crítico. La **facilidad de explotación** es alta si no hay validación de roles, y el **impacto sobre los datos** es potencialmente masivo.
2. **Inyección de código JavaScript (XSS)**.
    - **Justificación: Alta prioridad**. La vulnerabilidad es fácil de explotar y su impacto es directo y masivo, ya que afecta a todos los usuarios que visiten la página, comprometiendo sus sesiones y datos.

3. **Múltiples intentos fallidos de login**.
    - **Justificación: Prioridad alta-media**. Es un ataque en curso que puede ser un intento de fuerza bruta. Si tiene éxito, el **impacto sobre los usuarios y los datos** sería alto.
4. **Fallo en el backend que expone errores de BD**.
    - **Justificación: Prioridad media**. Aunque la vulnerabilidad ya existe, el **nivel de exposición** es alto, ya que los atacantes pueden usar esta información para explotar otras vulnerabilidades.
5. **Librería JavaScript con vulnerabilidad conocida**.
    - **Justificación: Prioridad baja**. Es el único evento que aún no ha causado un incidente de seguridad. La vulnerabilidad es un riesgo latente, pero el **impacto inmediato** es nulo.